Course Logistics and Introduction to Deep Learning

Deep Learning (DSE316/616)

Vinod K Kurmi
Assistant Professor, DSE

Indian Institute of Science Education and Research Bhopal

Aug 01, 2022

Course Logistics

- **Course name:** Deep Learning (DSE 316/616)
- **Timing and venue:** L3, Mon-Thur-6.00-7.30PM
- **Course website:** csgxcxo (slides/readings etc will be posted here)
- **discussion site:** csgxcxo (use it actively and responsibly)
- **Instructor:** Vinod K Kurmi (e-mail: vinodkk@iiserb.ac.in, office: Main Building-212B)
  - Prefix email subject by DSE316/616
  - Office Hours: Wed 5:00-6:30pm (by appointment)
- **Auditing:** Send me email to be added to the mailing list.
  - Will have access to all the course material; can participate in discussions
  - However, we are unable to grade your assignments/exams. Can't form project groups with creditors

Course Logistics

- **Course name:** Deep Learning (DSE 316/616)
- **Timing and venue:** L3, Mon-Thur-6.00-7.30PM
- **Course website:** csgxcxo (slides/readings etc will be posted here)
- **discussion site:** csgxcxo (use it actively and responsibly)
- **Instructor:** Vinod K Kurmi (e-mail: vinodkk@iiserb.ac.in, office: Main Building-212B)
  - Prefix email subject by DSE316/616
  - Office Hours: Wed 5:00-6:30pm (by appointment)
- **Auditing:** Send me email to be added to the mailing list.
  - Will have access to all the course material; can participate in discussions
  - However, we are unable to grade your assignments/exams. Can't form project groups with creditors

Course Logistics

- **Course name:** Deep Learning (DSE 316/616)
- **Timing and venue:** L3, Mon-Thur-6.00-7.30PM
- **Course website:** csgxcxo (slides/readings etc will be posted here)
- **discussion site:** csgxcxo (use it actively and responsibly)
- **Instructor:** Vinod K Kurmi (e-mail: vinodkk@iiserb.ac.in, office: Main Building-212B)
  - Prefix email subject by DSE316/616
  - Office Hours: Wed 5:00-6:30pm (by appointment)
- **Auditing:** Send me email to be added to the mailing list.
  - Will have access to all the course material; can participate in discussions
  - However, we are unable to grade your assignments/exams. Can't form project groups with creditors

Course Logistics

- **Course name:** Deep Learning (DSE 316/616)
- **Timing and venue:** L3, Mon-Thur-6.00-7.30PM
- **Course website:** csgxcxo (slides/readings etc will be posted here)
- **discussion site:** csgxcxo (use it actively and responsibly)
- **Instructor:** Vinod K Kurmi (e-mail: vinodkk@iiserb.ac.in, office: Main Building-212B)
  - Prefix email subject by DSE316/616
  - Office Hours: Wed 5:00-6:30pm (by appointment)
- **Auditing:** Send me email to be added to the mailing list.
  - Will have access to all the course material; can participate in discussions
  - However, we are unable to grade your assignments/exams. Can't form project groups with creditors

Course Logistics

- **Course name:** Deep Learning (DSE 316/616)
- **Timing and venue:** L3, Mon-Thur-6.00-7.30PM
- **Course website:** csgxcxo (slides/readings etc will be posted here)
- **discussion site:** csgxcxo (use it actively and responsibly)
- **Instructor:** Vinod K Kurmi (e-mail: vinodkk@iiserb.ac.in, office: Main Building-212B)
  - Prefix email subject by DSE316/616
  - Office Hours: Wed 5:00-6:30pm (by appointment)
- **Auditing:** Send me email to be added to the mailing list.
  - Will have access to all the course material; can participate in discussions
  - However, we are unable to grade your assignments/exams. Can't form project groups with creditors

Course Logistics

- **Course name:** Deep Learning (DSE 316/616)
- **Timing and venue:** L3, Mon-Thur-6.00-7.30PM
- **Course website:** csgxcxo (slides/readings etc will be posted here)
- **discussion site:** csgxcxo (use it actively and responsibly)
- **Instructor:** Vinod K Kurmi (e-mail: vinodkk@iiserb.ac.in, office: Main Building-212B)
  - Prefix email subject by DSE316/616
  - Office Hours: Wed 5:00-6:30pm (by appointment)
- **Auditing:** Send me email to be added to the mailing list.
  - Will have access to all the course material; can participate in discussions
  - However, we are unable to grade your assignments/exams. Can't form project groups with creditors

Course Logistics

- **Course name:** Deep Learning (DSE 316/616)
- **Timing and venue:** L3, Mon-Thur-6.00-7.30PM
- **Course website:** csgxcxo (slides/readings etc will be posted here)
- **discussion site:** csgxcxo (use it actively and responsibly)
- **Instructor:** Vinod K Kurmi (e-mail: vinodkk@iiserb.ac.in, office: Main Building-212B)
  - Prefix email subject by DSE316/616
  - Office Hours: Wed 5:00-6:30pm (by appointment)
- **Auditing:** Send me email to be added to the mailing list.
  - Will have access to all the course material; can participate in discussions
  - However, we are unable to grade your assignments/exams. Can't form project groups with creditors

Course Logistics

- **Course name:** Deep Learning (DSE 316/616)
- **Timing and venue:** L3, Mon-Thur-6.00-7.30PM
- **Course website:** csgxcxo (slides/readings etc will be posted here)
- **discussion site:** csgxcxo (use it actively and responsibly)
- **Instructor:** Vinod K Kurmi (e-mail: vinodkk@iiserb.ac.in, office: Main Building-212B)
  - Prefix email subject by DSE316/616
  - Office Hours: Wed 5:00-6:30pm (by appointment)
- **Auditing:** Send me email to be added to the mailing list.
  - Will have access to all the course material; can participate in discussions
  - However, we are unable to grade your assignments/exams. Can't form project groups with creditors

Course Logistics

- **Course name:** Deep Learning (DSE 316/616)
- **Timing and venue:** L3, Mon-Thur-6.00-7.30PM
- **Course website:** csgxcxo (slides/readings etc will be posted here)
- **discussion site:** csgxcxo (use it actively and responsibly)
- **Instructor:** Vinod K Kurmi (e-mail: vinodkk@iiserb.ac.in, office: Main Building-212B)
    - Prefix email subject by DSE316/616
    - Office Hours: Wed 5:00-6:30pm (by appointment)
- **Auditing:** Send me email to be added to the mailing list.
    - Will have access to all the course material; can participate in discussions
    - However, we are unable to grade your assignments/exams. Can't form project groups with creditors

Course Logistics

- **Course name:** Deep Learning (DSE 316/616)
- **Timing and venue:** L3, Mon-Thur-6.00-7.30PM
- **Course website:** csgxcxo (slides/readings etc will be posted here)
- **discussion site:** csgxcxo (use it actively and responsibly)
- **Instructor:** Vinod K Kurmi (e-mail: vinodkk@iiserb.ac.in, office: Main Building-212B)
  - Prefix email subject by DSE316/616
  - Office Hours: Wed 5:00-6:30pm (by appointment)
- **Auditing:** Send me email to be added to the mailing list.
  - Will have access to all the course material; can participate in discussions
  - However, we are unable to grade your assignments/exams. Can't form project groups with creditors

Grading Scheme

- 2-3 assignments and attendance:
    - Written questions + some programming in Python/MATLAB
- 2-3 quizzes:
- 2 exams:
    - Midterm exam:
    - Final exam:
- Class Project
    - Research project, to be done in groups of 3
    - More details will be shared very soon

Grading Scheme

- 2-3 assignments and attendance:
  - Written questions + some programming in Python/MATLAB
- 2-3 quizzes:
- 2 exams:
  - Midterm exam:
  - Final exam:
- Class Project
  - Research project, to be done in groups of 3
  - More details will be shared very soon

Grading Scheme

- 2-3 assignments and attendance:
  - Written questions + some programming in Python/MATLAB
- 2-3 quizzes:
- 2 exams:
  - Midterm exam:
  - Final exam:
- Class Project
  - Research project, to be done in groups of 3
  - More details will be shared very soon

Grading Scheme

- 2-3 assignments and attendance:
  - Written questions + some programming in Python/MATLAB
- 2-3 quizzes:
- 2 exams:
  - Midterm exam:
  - Final exam:
- Class Project
  - Research project, to be done in groups of 3
  - More details will be shared very soon

Grading Scheme

- 2-3 assignments and attendance:
  - Written questions + some programming in Python/MATLAB
- 2-3 quizzes:
- 2 exams:
  - Midterm exam:
  - Final exam:
- Class Project
  - Research project, to be done in groups of 3
  - More details will be shared very soon

Grading Scheme

- 2-3 assignments and attendance:
  - Written questions + some programming in Python/MATLAB
- 2-3 quizzes:
- 2 exams:
  - Midterm exam:
  - Final exam:
- Class Project
  - Research project, to be done in groups of 3
  - More details will be shared very soon

Grading Scheme

- 2-3 assignments and attendance:
  - Written questions + some programming in Python/MATLAB
- 2-3 quizzes:
- 2 exams:
  - Midterm exam:
  - Final exam:
- Class Project
  - Research project, to be done in groups of 3
  - More details will be shared very soon

Grading Scheme

- 2-3 assignments and attendance:
  - Written questions + some programming in Python/MATLAB
- 2-3 quizzes:
- 2 exams:
  - Midterm exam:
  - Final exam:
- Class Project
  - Research project, to be done in groups of 3
  - More details will be shared very soon

Grading Scheme

- 2-3 assignments and attendance:
  - Written questions + some programming in Python/MATLAB
- 2-3 quizzes:
- 2 exams:
  - Midterm exam:
  - Final exam:
- Class Project
  - Research project, to be done in groups of 3
  - More details will be shared very soon

Collaboration vs Cheating

- Collaboration is encouraged. Cheating/copying will lead to strict punishments.
- Feel free to discuss homework assignments with your classmates.
- Must write your own solution in your own words (same goes for coding assignments) Plagiarism from other sources (for assignments/project) will also lead to strict punishment
- Other things that will lead to punishment Use of unfair means in the exams
- Fabricating experimental results in assignments/project Important: Both copying as well as helping someone copy will be equally punishable

Collaboration vs Cheating

- Collaboration is encouraged. Cheating/copying will lead to strict punishments.
- Feel free to discuss homework assignments with your classmates.
- Must write your own solution in your own words (same goes for coding assignments) Plagiarism from other sources (for assignments/project) will also lead to strict punishment
- Other things that will lead to punishment Use of unfair means in the exams
- Fabricating experimental results in assignments/project Important: Both copying as well as helping someone copy will be equally punishable

Collaboration vs Cheating

- Collaboration is encouraged. Cheating/copying will lead to strict punishments.
- Feel free to discuss homework assignments with your classmates.
- Must write your own solution in your own words (same goes for coding assignments) Plagiarism from other sources (for assignments/project) will also lead to strict punishment
- Other things that will lead to punishment Use of unfair means in the exams
- Fabricating experimental results in assignments/project Important: Both copying as well as helping someone copy will be equally punishable

Collaboration vs Cheating

- Collaboration is encouraged. Cheating/copying will lead to strict punishments.
- Feel free to discuss homework assignments with your classmates.
- Must write your own solution in your own words (same goes for coding assignments) Plagiarism from other sources (for assignments/project) will also lead to strict punishment
- Other things that will lead to punishment Use of unfair means in the exams
- Fabricating experimental results in assignments/project Important: Both copying as well as helping someone copy will be equally punishable

Collaboration vs Cheating

- Collaboration is encouraged. Cheating/copying will lead to strict punishments.
- Feel free to discuss homework assignments with your classmates.
- Must write your own solution in your own words (same goes for coding assignments) Plagiarism from other sources (for assignments/project) will also lead to strict punishment
- Other things that will lead to punishment Use of unfair means in the exams
- Fabricating experimental results in assignments/project Important: Both copying as well as helping someone copy will be equally punishable

Course Policies

- Repeat: Absolutely ZERO tolerance for cheating
  - Punishable as per institute's/department's rules
- Requests for homework extensions won't be entertained
  - Can submit homeworks upto 3 late days with 10% penalty per day
  - Every student entitled for ONE late homework submission without penalty (use it wisely)
- Use Piazza actively and responsibly
  - Limited to discussions related to class
  - Allowed to remain anonymous to classmates but not to instructors
  - Avoid asking questions privately (so that everyone can benefit from the question/answer)
  - Questions should not be attempts to get/verify answers to homework problems

Course Policies

- Repeat: Absolutely ZERO tolerance for cheating
  - Punishable as per institute's/department's rules
- Requests for homework extensions won't be entertained
  - Can submit homeworks upto 3 late days with 10% penalty per day
  - Every student entitled for ONE late homework submission without penalty (use it wisely)
- Use Piazza actively and responsibly
  - Limited to discussions related to class
  - Allowed to remain anonymous to classmates but not to instructors
  - Avoid asking questions privately (so that everyone can benefit from the question/answer)
  - Questions should not be attempts to get/verify answers to homework problems

Course Policies

- Repeat: Absolutely ZERO tolerance for cheating
  - Punishable as per institute's/department's rules
- Requests for homework extensions won't be entertained
  - Can submit homeworks upto 3 late days with 10% penalty per day
  - Every student entitled for ONE late homework submission without penalty (use it wisely)
- Use Piazza actively and responsibly
  - Limited to discussions related to class
  - Allowed to remain anonymous to classmates but not to instructors
  - Avoid asking questions privately (so that everyone can benefit from the question/answer)
  - Questions should not be attempts to get/verify answers to homework problems

Course Policies

- Repeat: Absolutely ZERO tolerance for cheating
  - Punishable as per institute's/department's rules
- Requests for homework extensions won't be entertained
  - Can submit homeworks upto 3 late days with 10% penalty per day
  - Every student entitled for ONE late homework submission without penalty (use it wisely)
- Use Piazza actively and responsibly
  - Limited to discussions related to class
  - Allowed to remain anonymous to classmates but not to instructors
  - Avoid asking questions privately (so that everyone can benefit from the question/answer)
  - Questions should not be attempts to get/verify answers to homework problems

Course Policies

- Repeat: Absolutely ZERO tolerance for cheating
  - Punishable as per institute's/department's rules
- Requests for homework extensions won't be entertained
  - Can submit homeworks upto 3 late days with 10% penalty per day
  - Every student entitled for ONE late homework submission without penalty (use it wisely)
- Use Piazza actively and responsibly
  - Limited to discussions related to class
  - Allowed to remain anonymous to classmates but not to instructors
  - Avoid asking questions privately (so that everyone can benefit from the question/answer)
  - Questions should not be attempts to get/verify answers to homework problems

Course Policies

- Repeat: Absolutely ZERO tolerance for cheating
  - Punishable as per institute's/department's rules
- Requests for homework extensions won't be entertained
  - Can submit homeworks upto 3 late days with 10% penalty per day
  - Every student entitled for ONE late homework submission without penalty (use it wisely)
- Use Piazza actively and responsibly
  - Limited to discussions related to class
  - Allowed to remain anonymous to classmates but not to instructors
  - Avoid asking questions privately (so that everyone can benefit from the question/answer)
  - Questions should not be attempts to get/verify answers to homework problems

Course Policies

- Repeat: Absolutely ZERO tolerance for cheating
  - Punishable as per institute's/department's rules
- Requests for homework extensions won't be entertained
  - Can submit homeworks upto 3 late days with 10% penalty per day
  - Every student entitled for ONE late homework submission without penalty (use it wisely)
- Use Piazza actively and responsibly
  - Limited to discussions related to class
  - Allowed to remain anonymous to classmates but not to instructors
  - Avoid asking questions privately (so that everyone can benefit from the question/answer)
  - Questions should not be attempts to get/verify answers to homework problems

Course Policies

- Repeat: Absolutely ZERO tolerance for cheating
  - Punishable as per institute's/department's rules
- Requests for homework extensions won't be entertained
  - Can submit homeworks upto 3 late days with 10% penalty per day
  - Every student entitled for ONE late homework submission without penalty (use it wisely)
- Use Piazza actively and responsibly
  - Limited to discussions related to class
  - Allowed to remain anonymous to classmates but not to instructors
  - Avoid asking questions privately (so that everyone can benefit from the question/answer)
  - Questions should not be attempts to get/verify answers to homework problems

Course Policies

- Repeat: Absolutely ZERO tolerance for cheating
  - Punishable as per institute's/department's rules
- Requests for homework extensions won't be entertained
  - Can submit homeworks upto 3 late days with 10% penalty per day
  - Every student entitled for ONE late homework submission without penalty (use it wisely)
- Use Piazza actively and responsibly
  - Limited to discussions related to class
  - Allowed to remain anonymous to classmates but not to instructors
  - Avoid asking questions privately (so that everyone can benefit from the question/answer)
  - Questions should not be attempts to get/verify answers to homework problems

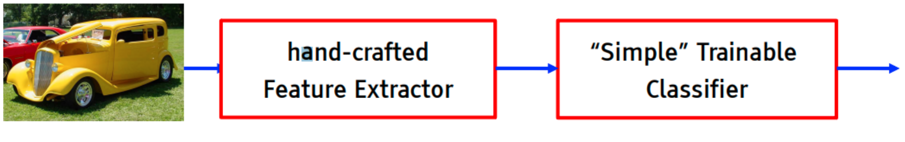
Course Policies

- Repeat: Absolutely ZERO tolerance for cheating
  - Punishable as per institute's/department's rules
- Requests for homework extensions won't be entertained
  - Can submit homeworks upto 3 late days with 10% penalty per day
  - Every student entitled for ONE late homework submission without penalty (use it wisely)
- Use Piazza actively and responsibly
  - Limited to discussions related to class
  - Allowed to remain anonymous to classmates but not to instructors
  - Avoid asking questions privately (so that everyone can benefit from the question/answer)
  - Questions should not be attempts to get/verify answers to homework problems

Deep Learning = Learning Representations/Features

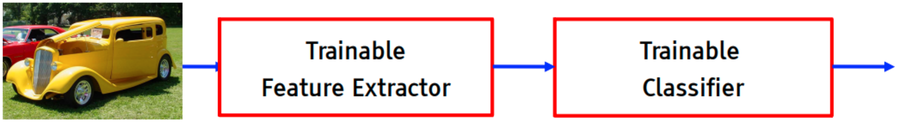
The traditional model of pattern recognition (since the late 50's)

Fixed/engineered features (or fixed kernel) + trainable classifier



End-to-end learning / Feature learning / Deep learning

Trainable features (or kernel) + trainable classifier



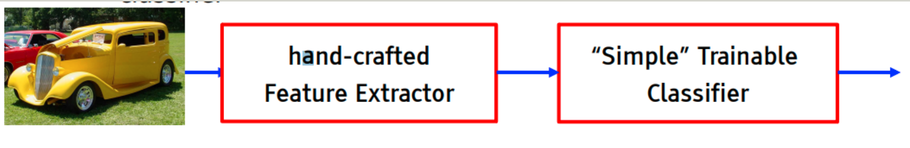
[1]

[1] Slides Credit: Y LeCun

Deep Learning = Learning Representations/Features

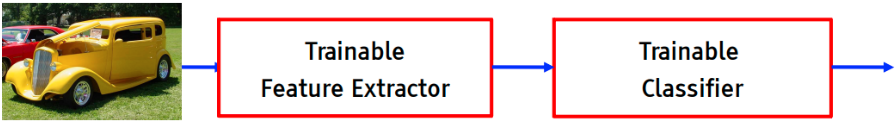
The traditional model of pattern recognition (since the late 50's)

Fixed/engineered features (or fixed kernel) + trainable classifier



End-to-end learning / Feature learning / Deep learning

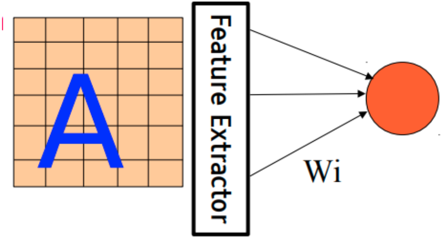
Trainable features (or kernel) + trainable classifier



[1]

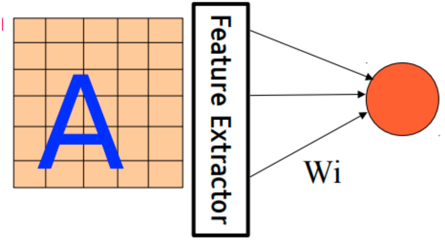
[1] Slides Credit: Y LeCun

History



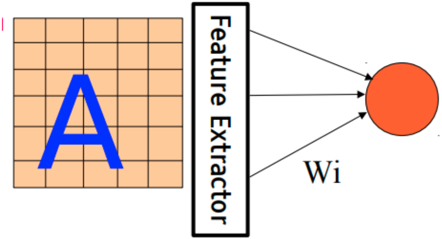
- The first learning machine: the Perceptron
  - Built at Cornell in 1960
- The Perceptron was a linear classifier on top of a simple feature extractor
- The vast majority of practical applications of ML today use glorified linear classifiers or glorified template matching
- Designing a feature extractor requires considerable efforts by experts

History



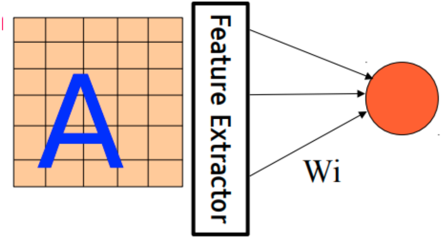
- The first learning machine: the Perceptron
  - Built at Cornell in 1960
- The Perceptron was a linear classifier on top of a simple feature extractor
- The vast majority of practical applications of ML today use glorified linear classifiers or glorified template matching
- Designing a feature extractor requires considerable efforts by experts

History



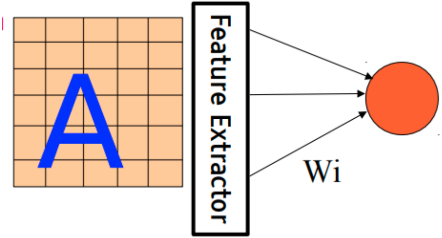
- The first learning machine: the Perceptron
  - Built at Cornell in 1960
- The Perceptron was a linear classifier on top of a simple feature extractor
- The vast majority of practical applications of ML today use glorified linear classifiers or glorified template matching
- Designing a feature extractor requires considerable efforts by experts

History



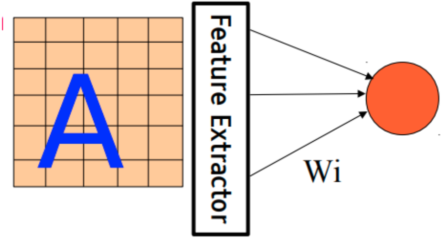
- The first learning machine: the Perceptron
  - Built at Cornell in 1960
- The Perceptron was a linear classifier on top of a simple feature extractor
- The vast majority of practical applications of ML today use glorified linear classifiers or glorified template matching
- Designing a feature extractor requires considerable efforts by experts

History



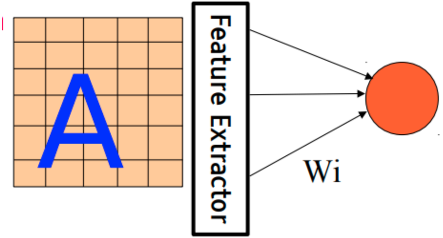
- The first learning machine: the Perceptron
  - Built at Cornell in 1960
- The Perceptron was a linear classifier on top of a simple feature extractor
- The vast majority of practical applications of ML today use glorified linear classifiers or glorified template matching
- Designing a feature extractor requires considerable efforts by experts

History



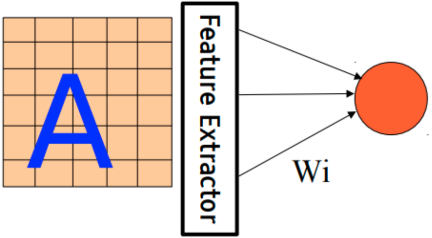
- The first learning machine: the Perceptron
  - Built at Cornell in 1960
- The Perceptron was a linear classifier on top of a simple feature extractor
- The vast majority of practical applications of ML today use glorified linear classifiers or glorified template matching
- Designing a feature extractor requires considerable efforts by experts

Linear Machines And their limitations



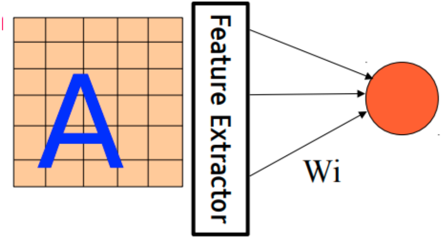
- The first learning machine: the Perceptron
  - Built at Cornell in 1960
- The Perceptron was a linear classifier on top of a simple feature extractor
- The vast majority of practical applications of ML today use glorified linear classifiers or glorified template matching
- Designing a feature extractor requires considerable efforts by experts

Linear Machines And their limitations



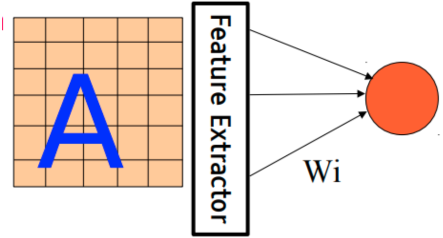
- The first learning machine: the Perceptron
  - Built at Cornell in 1960
- The Perceptron was a linear classifier on top of a simple feature extractor
- The vast majority of practical applications of ML today use glorified linear classifiers or glorified template matching
- Designing a feature extractor requires considerable efforts by experts

Linear Machines And their limitations



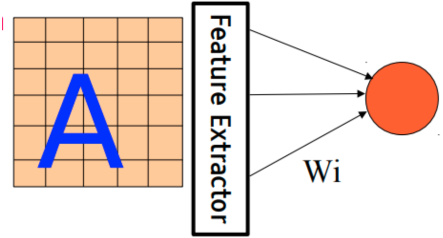
- The first learning machine: the Perceptron
  - Built at Cornell in 1960
- The Perceptron was a linear classifier on top of a simple feature extractor
- The vast majority of practical applications of ML today use glorified linear classifiers or glorified template matching
- Designing a feature extractor requires considerable efforts by experts

Linear Machines And their limitations



- The first learning machine: the Perceptron
  - Built at Cornell in 1960
- The Perceptron was a linear classifier on top of a simple feature extractor
- The vast majority of practical applications of ML today use glorified linear classifiers or glorified template matching
- Designing a feature extractor requires considerable efforts by experts

Linear Machines And their limitations



- The first learning machine: the Perceptron
  - Built at Cornell in 1960
- The Perceptron was a linear classifier on top of a simple feature extractor
- The vast majority of practical applications of ML today use glorified linear classifiers or glorified template matching
- Designing a feature extractor requires considerable efforts by experts

Linear Machines and their limitations

Linear Regression, Mean Square Loss:

- decision rule: $y = W'X$
- loss function: $L(W, y^i, X^i) = \frac{1}{2}(y^i - W'X^i)^2$
- gradient of loss: $\frac{\partial L(W, y^i, X^i)}{\partial W}' = -(y^i - W(t)'X^i)X^i$
- update rule: $W(t+1) = W(t) + \eta(t)(y^i - W(t)'X^i)X^i$
- direct solution: solve linear system $[\sum_{i=1}^{P} X^i X^{i\prime}]W = \sum_{i=1}^{P} y^i X^i$

Linear Machines and their limitations

Perceptron:

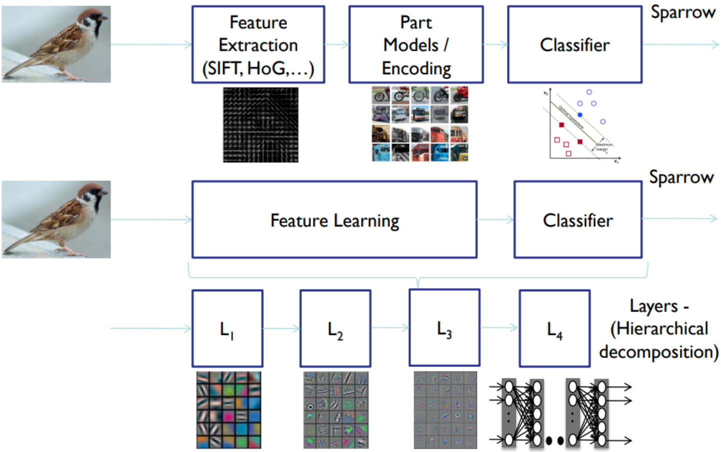
- decision rule: $y = F(W'X)$ (F is the threshold function)
- loss function: $L(W, y^i, X^i) = (F(W'X^i) - y^i)W'X^i$
- gradient of loss: $\frac{\partial L(W,y^i,X^i)}{\partial W}' = -(y^i - F(W(t)'X^i))X^i$
- update rule: $W(t+1) = W(t) + \eta(t)(y^i - F(W(t)'X^i))X^i$
- direct solution: find W such that $-y^i F(W'X^i) < 0 \quad \forall i$

Linear Machines and their limitations

Logistic Regression, Negative Log-Likelihood Loss function:

- decision rule: $y = F(W'X)$, with $F(a) = \tanh(a) = \frac{1-\exp(a)}{1+\exp(a)}$ (sigmoid function).
- loss function: $L(W, y^i, X^i) = 2\log(1 + \exp(-y^i W' X^i))$
- gradient of loss: $\frac{\partial L(W, y^i, X^i)}{\partial W}' = -\left(Y^i - F(W'X)\right) X^i$
- update rule: $W(t+1) = W(t) + \eta(t)(y^i - F(W(t)'X^i))X^i$

Paradigm Shift

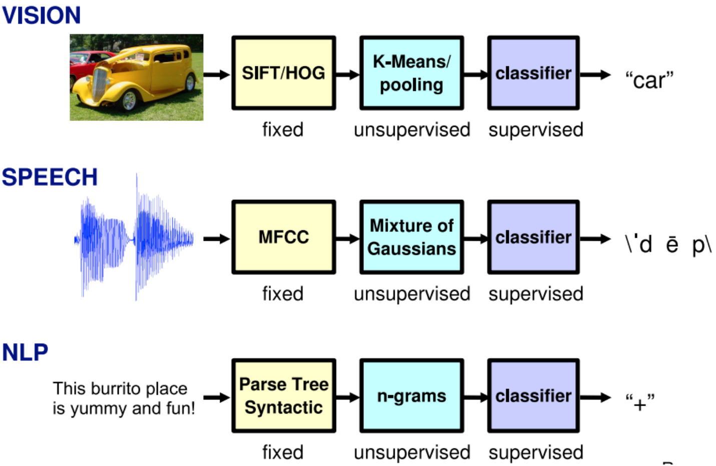


2

---

[2] Slides Credit: CV Jawahar

Common pipeline



Abstract pipeline

VISION

pixels → edge → texton → motif → part → object

SPEECH

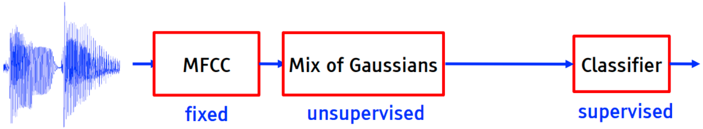
sample → spectral band → formant → motif → phone → word

NLP

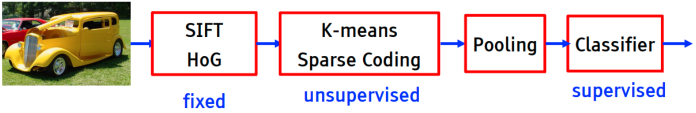
character → word → NP/VP/.. → clause → sentence → story

Learning Hierarchical Representations

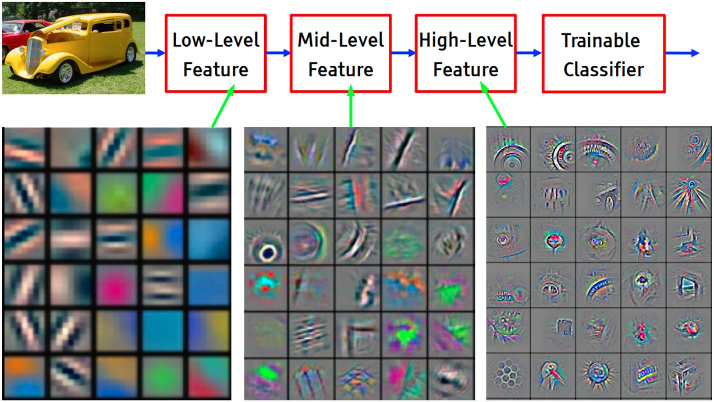
Speech recognition: early 90's – 2011



Object Recognition: 2006 - 2012



Learning Hierarchical Representations



Learning Representations: a challenge for ML, CV, AI, Neuroscience, Cognitive Science...

- How do we learn representations of the perceptual world?
  - How can a perceptual system build itself by looking at the world?
  - How much prior structure is necessary
- ML/AI: how do we learn features or feature hierarchies?
  - What is the fundamental principle? What is the learning algorithm? What is the architecture?
- Neuroscience: how does the cortex learn perception?
  - Does the cortex "run" a single, general learning algorithm? (or a small number of them)
- CogSci: how does the mind learn abstract concepts on top of less abstract ones?
- Deep Learning addresses the problem of learning hierarchical representations with a single algorithm.

Learning Representations: a challenge for ML, CV, AI, Neuroscience, Cognitive Science...

- How do we learn representations of the perceptual world?
  - How can a perceptual system build itself by looking at the world?
  - How much prior structure is necessary
- ML/AI: how do we learn features or feature hierarchies?
  - What is the fundamental principle? What is the learning algorithm? What is the architecture?
- Neuroscience: how does the cortex learn perception?
  - Does the cortex “run” a single, general learning algorithm? (or a small number of them)
- CogSci: how does the mind learn abstract concepts on top of less abstract ones?
- Deep Learning addresses the problem of learning hierarchical representations with a single algorithm.

Learning Representations: a challenge for ML, CV, AI, Neuroscience, Cognitive Science...

- How do we learn representations of the perceptual world?
  - How can a perceptual system build itself by looking at the world?
  - How much prior structure is necessary
- ML/AI: how do we learn features or feature hierarchies?
  - What is the fundamental principle? What is the learning algorithm? What is the architecture?
- Neuroscience: how does the cortex learn perception?
  - Does the cortex "run" a single, general learning algorithm? (or a small number of them)
- CogSci: how does the mind learn abstract concepts on top of less abstract ones?
- Deep Learning addresses the problem of learning hierarchical representations with a single algorithm.

Learning Representations: a challenge for ML, CV, AI, Neuroscience, Cognitive Science...

- How do we learn representations of the perceptual world?
  - How can a perceptual system build itself by looking at the world?
  - How much prior structure is necessary
- ML/AI: how do we learn features or feature hierarchies?
  - What is the fundamental principle? What is the learning algorithm? What is the architecture?
- Neuroscience: how does the cortex learn perception?
  - Does the cortex "run" a single, general learning algorithm? (or a small number of them)
- CogSci: how does the mind learn abstract concepts on top of less abstract ones?
- Deep Learning addresses the problem of learning hierarchical representations with a single algorithm.

Learning Representations: a challenge for ML, CV, AI, Neuroscience, Cognitive Science...

- How do we learn representations of the perceptual world?
  - How can a perceptual system build itself by looking at the world?
  - How much prior structure is necessary
- ML/AI: how do we learn features or feature hierarchies?
  - What is the fundamental principle? What is the learning algorithm? What is the architecture?
- Neuroscience: how does the cortex learn perception?
  - Does the cortex "run" a single, general learning algorithm? (or a small number of them)
- CogSci: how does the mind learn abstract concepts on top of less abstract ones?
- Deep Learning addresses the problem of learning hierarchical representations with a single algorithm.

Learning Representations: a challenge for ML, CV, AI, Neuroscience, Cognitive Science...

- How do we learn representations of the perceptual world?
  - How can a perceptual system build itself by looking at the world?
  - How much prior structure is necessary
- ML/AI: how do we learn features or feature hierarchies?
  - What is the fundamental principle? What is the learning algorithm? What is the architecture?
- Neuroscience: how does the cortex learn perception?
  - Does the cortex "run" a single, general learning algorithm? (or a small number of them)
- CogSci: how does the mind learn abstract concepts on top of less abstract ones?
- Deep Learning addresses the problem of learning hierarchical representations with a single algorithm.

Learning Representations: a challenge for ML, CV, AI, Neuroscience, Cognitive Science...

- How do we learn representations of the perceptual world?
  - How can a perceptual system build itself by looking at the world?
  - How much prior structure is necessary
- ML/AI: how do we learn features or feature hierarchies?
  - What is the fundamental principle? What is the learning algorithm? What is the architecture?
- Neuroscience: how does the cortex learn perception?
  - Does the cortex "run" a single, general learning algorithm? (or a small number of them)
- CogSci: how does the mind learn abstract concepts on top of less abstract ones?
- Deep Learning addresses the problem of learning hierarchical representations with a single algorithm.

Learning Representations: a challenge for ML, CV, AI, Neuroscience, Cognitive Science...

- How do we learn representations of the perceptual world?
  - How can a perceptual system build itself by looking at the world?
  - How much prior structure is necessary
- ML/AI: how do we learn features or feature hierarchies?
  - What is the fundamental principle? What is the learning algorithm? What is the architecture?
- Neuroscience: how does the cortex learn perception?
  - Does the cortex "run" a single, general learning algorithm? (or a small number of them)
- CogSci: how does the mind learn abstract concepts on top of less abstract ones?
- Deep Learning addresses the problem of learning hierarchical representations with a single algorithm.

Learning Representations: a challenge for ML, CV, AI, Neuroscience, Cognitive Science...

- How do we learn representations of the perceptual world?
  - How can a perceptual system build itself by looking at the world?
  - How much prior structure is necessary
- ML/AI: how do we learn features or feature hierarchies?
  - What is the fundamental principle? What is the learning algorithm? What is the architecture?
- Neuroscience: how does the cortex learn perception?
  - Does the cortex “run” a single, general learning algorithm? (or a small number of them)
- CogSci: how does the mind learn abstract concepts on top of less abstract ones?
- Deep Learning addresses the problem of learning hierarchical representations with a single algorithm.

Types of Training Protocols

- Purely Supervised
  - Initialize parameters randomly
  - Train in supervised mode
    - typically with SGD, using backprop to compute gradients
  - Used in most practical systems for speech and image recognition
- Unsupervised, layerwise + supervised classifier on top
  - Train each layer unsupervised, one after the other
  - Train a supervised classifier on top, keeping the other layers fixed
  - Good when very few labeled samples are available
- Unsupervised, layerwise + global supervised fine-tuning
  - Train each layer unsupervised, one after the other
  - Add a classifier layer, and retrain the whole thing supervised
  - Good when label set is poor (e.g. pedestrian detection)

Types of Training Protocols

- Purely Supervised
  - Initialize parameters randomly
  - Train in supervised mode
    - typically with SGD, using backprop to compute gradients
  - Used in most practical systems for speech and image recognition
- Unsupervised, layerwise + supervised classifier on top
  - Train each layer unsupervised, one after the other
  - Train a supervised classifier on top, keeping the other layers fixed
  - Good when very few labeled samples are available
- Unsupervised, layerwise + global supervised fine-tuning
  - Train each layer unsupervised, one after the other
  - Add a classifier layer, and retrain the whole thing supervised
  - Good when label set is poor (e.g. pedestrian detection)

Types of Training Protocols

- Purely Supervised
  - Initialize parameters randomly
  - Train in supervised mode
    - typically with SGD, using backprop to compute gradients
  - Used in most practical systems for speech and image recognition
- Unsupervised, layerwise + supervised classifier on top
  - Train each layer unsupervised, one after the other
  - Train a supervised classifier on top, keeping the other layers fixed
  - Good when very few labeled samples are available
- Unsupervised, layerwise + global supervised fine-tuning
  - Train each layer unsupervised, one after the other
  - Add a classifier layer, and retrain the whole thing supervised
  - Good when label set is poor (e.g. pedestrian detection)

Types of Training Protocols

- Purely Supervised
  - Initialize parameters randomly
  - Train in supervised mode
    - typically with SGD, using backprop to compute gradients
  - Used in most practical systems for speech and image recognition
- Unsupervised, layerwise + supervised classifier on top
  - Train each layer unsupervised, one after the other
  - Train a supervised classifier on top, keeping the other layers fixed
  - Good when very few labeled samples are available
- Unsupervised, layerwise + global supervised fine-tuning
  - Train each layer unsupervised, one after the other
  - Add a classifier layer, and retrain the whole thing supervised
  - Good when label set is poor (e.g. pedestrian detection)

Types of Training Protocols

- Purely Supervised
  - Initialize parameters randomly
  - Train in supervised mode
    - typically with SGD, using backprop to compute gradients
  - Used in most practical systems for speech and image recognition
- Unsupervised, layerwise + supervised classifier on top
  - Train each layer unsupervised, one after the other
  - Train a supervised classifier on top, keeping the other layers fixed
  - Good when very few labeled samples are available
- Unsupervised, layerwise + global supervised fine-tuning
  - Train each layer unsupervised, one after the other
  - Add a classifier layer, and retrain the whole thing supervised
  - Good when label set is poor (e.g. pedestrian detection)

Types of Training Protocols

- Purely Supervised
  - Initialize parameters randomly
  - Train in supervised mode
    - typically with SGD, using backprop to compute gradients
  - Used in most practical systems for speech and image recognition
- Unsupervised, layerwise + supervised classifier on top
  - Train each layer unsupervised, one after the other
  - Train a supervised classifier on top, keeping the other layers fixed
  - Good when very few labeled samples are available
- Unsupervised, layerwise + global supervised fine-tuning
  - Train each layer unsupervised, one after the other
  - Add a classifier layer, and retrain the whole thing supervised
  - Good when label set is poor (e.g. pedestrian detection)

Types of Training Protocols

- Purely Supervised
  - Initialize parameters randomly
  - Train in supervised mode
    - typically with SGD, using backprop to compute gradients
  - Used in most practical systems for speech and image recognition
- Unsupervised, layerwise + supervised classifier on top
  - Train each layer unsupervised, one after the other
  - Train a supervised classifier on top, keeping the other layers fixed
  - Good when very few labeled samples are available
- Unsupervised, layerwise + global supervised fine-tuning
  - Train each layer unsupervised, one after the other
  - Add a classifier layer, and retrain the whole thing supervised
  - Good when label set is poor (e.g. pedestrian detection)

Types of Training Protocols

- Purely Supervised
  - Initialize parameters randomly
  - Train in supervised mode
    - typically with SGD, using backprop to compute gradients
  - Used in most practical systems for speech and image recognition
- Unsupervised, layerwise + supervised classifier on top
  - Train each layer unsupervised, one after the other
  - Train a supervised classifier on top, keeping the other layers fixed
  - Good when very few labeled samples are available
- Unsupervised, layerwise + global supervised fine-tuning
  - Train each layer unsupervised, one after the other
  - Add a classifier layer, and retrain the whole thing supervised
  - Good when label set is poor (e.g. pedestrian detection)

Types of Training Protocols

- Purely Supervised
  - Initialize parameters randomly
  - Train in supervised mode
    - typically with SGD, using backprop to compute gradients
  - Used in most practical systems for speech and image recognition
- Unsupervised, layerwise + supervised classifier on top
  - Train each layer unsupervised, one after the other
  - Train a supervised classifier on top, keeping the other layers fixed
  - Good when very few labeled samples are available
- Unsupervised, layerwise + global supervised fine-tuning
  - Train each layer unsupervised, one after the other
  - Add a classifier layer, and retrain the whole thing supervised
  - Good when label set is poor (e.g. pedestrian detection)

Types of Training Protocols

- Purely Supervised
  - Initialize parameters randomly
  - Train in supervised mode
    - typically with SGD, using backprop to compute gradients
  - Used in most practical systems for speech and image recognition
- Unsupervised, layerwise + supervised classifier on top
  - Train each layer unsupervised, one after the other
  - Train a supervised classifier on top, keeping the other layers fixed
  - Good when very few labeled samples are available
- Unsupervised, layerwise + global supervised fine-tuning
  - Train each layer unsupervised, one after the other
  - Add a classifier layer, and retrain the whole thing supervised
  - Good when label set is poor (e.g. pedestrian detection)

Types of Training Protocols

- Purely Supervised
  - Initialize parameters randomly
  - Train in supervised mode
    - typically with SGD, using backprop to compute gradients
  - Used in most practical systems for speech and image recognition
- Unsupervised, layerwise + supervised classifier on top
  - Train each layer unsupervised, one after the other
  - Train a supervised classifier on top, keeping the other layers fixed
  - Good when very few labeled samples are available
- Unsupervised, layerwise + global supervised fine-tuning
  - Train each layer unsupervised, one after the other
  - Add a classifier layer, and retrain the whole thing supervised
  - Good when label set is poor (e.g. pedestrian detection)

Types of Training Protocols

- Purely Supervised
  - Initialize parameters randomly
  - Train in supervised mode
    - typically with SGD, using backprop to compute gradients
  - Used in most practical systems for speech and image recognition
- Unsupervised, layerwise + supervised classifier on top
  - Train each layer unsupervised, one after the other
  - Train a supervised classifier on top, keeping the other layers fixed
  - Good when very few labeled samples are available
- Unsupervised, layerwise + global supervised fine-tuning
  - Train each layer unsupervised, one after the other
  - Add a classifier layer, and retrain the whole thing supervised
  - Good when label set is poor (e.g. pedestrian detection)

Types of Training Protocols

- Purely Supervised
  - Initialize parameters randomly
  - Train in supervised mode
    - typically with SGD, using backprop to compute gradients
  - Used in most practical systems for speech and image recognition
- Unsupervised, layerwise + supervised classifier on top
  - Train each layer unsupervised, one after the other
  - Train a supervised classifier on top, keeping the other layers fixed
  - Good when very few labeled samples are available
- Unsupervised, layerwise + global supervised fine-tuning
  - Train each layer unsupervised, one after the other
  - Add a classifier layer, and retrain the whole thing supervised
  - Good when label set is poor (e.g. pedestrian detection)

Why would deep architectures be more efficient?

- A deep architecture trades space for time (or breadth for depth)
  - more layers (more sequential computation),
  - but less hardware (less parallel computation).

Why would deep architectures be more efficient?

- A deep architecture trades space for time (or breadth for depth)
  - more layers (more sequential computation),
  - but less hardware (less parallel computation).

Why would deep architectures be more efficient?

- A deep architecture trades space for time (or breadth for depth)
  - more layers (more sequential computation),
  - but less hardware (less parallel computation).

Which Models are Deep?

- 2-layer models are not deep (even if you train the first layer)
  - Because there is no feature hierarchy
- Neural nets with 1 hidden layer are not deep
- SVMs and Kernel methods are not deep
- Classification trees are not deep
- Graphical models can be deep (but most are not)

Which Models are Deep?

- 2-layer models are not deep (even if you train the first layer)
  - Because there is no feature hierarchy
- Neural nets with 1 hidden layer are not deep
- SVMs and Kernel methods are not deep
- Classification trees are not deep
- Graphical models can be deep (but most are not)

Which Models are Deep?

- 2-layer models are not deep (even if you train the first layer)
  - Because there is no feature hierarchy
- Neural nets with 1 hidden layer are not deep
- SVMs and Kernel methods are not deep
- Classification trees are not deep
- Graphical models can be deep (but most are not)

Which Models are Deep?

- 2-layer models are not deep (even if you train the first layer)
  - Because there is no feature hierarchy
- Neural nets with 1 hidden layer are not deep
- SVMs and Kernel methods are not deep
- Classification trees are not deep
- Graphical models can be deep (but most are not)

Which Models are Deep?

- 2-layer models are not deep (even if you train the first layer)
  - Because there is no feature hierarchy
- Neural nets with 1 hidden layer are not deep
- SVMs and Kernel methods are not deep
- Classification trees are not deep
- Graphical models can be deep (but most are not)

Which Models are Deep?

- 2-layer models are not deep (even if you train the first layer)
    - Because there is no feature hierarchy
- Neural nets with 1 hidden layer are not deep
- SVMs and Kernel methods are not deep
- Classification trees are not deep
- Graphical models can be deep (but most are not)

Deep Learning: A Theoretician's Nightmare?

- Deep Learning involves non-convex loss functions
  - With non-convex losses, all bets are off
  - Then again, every speech recognition system ever deployed has used non-convex optimization (GMMs are non convex). hierarchy
- No generalization bounds?
  - Actually, the usual VC bounds apply: most deep learning systems have a finite VC dimension
  - We don't have tighter bounds than that.
  - But then again, how many bounds are tight enough to be useful for model selection?
- It's hard to prove anything about deep learning systems
  - Then again, if we only study models for which we can prove things, we wouldn't have speech, handwriting, and visual object recognition systems today

Deep Learning: A Theoretician's Nightmare?

- Deep Learning involves non-convex loss functions
  - With non-convex losses, all bets are off
  - Then again, every speech recognition system ever deployed has used non-convex optimization (GMMs are non convex). hierarchy
- No generalization bounds?
  - Actually, the usual VC bounds apply: most deep learning systems have a finite VC dimension
  - We don't have tighter bounds than that.
  - But then again, how many bounds are tight enough to be useful for model selection?
- It's hard to prove anything about deep learning systems
  - Then again, if we only study models for which we can prove things, we wouldn't have speech, handwriting, and visual object recognition systems today

Deep Learning: A Theoretician's Nightmare?

- Deep Learning involves non-convex loss functions
  - With non-convex losses, all bets are off
  - Then again, every speech recognition system ever deployed has used non-convex optimization (GMMs are non convex). hierarchy
- No generalization bounds?
  - Actually, the usual VC bounds apply: most deep learning systems have a finite VC dimension
  - We don't have tighter bounds than that.
  - But then again, how many bounds are tight enough to be useful for model selection?
- It's hard to prove anything about deep learning systems
  - Then again, if we only study models for which we can prove things, we wouldn't have speech, handwriting, and visual object recognition systems today

Deep Learning: A Theoretician's Nightmare?

- Deep Learning involves non-convex loss functions
  - With non-convex losses, all bets are off
  - Then again, every speech recognition system ever deployed has used non-convex optimization (GMMs are non convex). hierarchy
- No generalization bounds?
  - Actually, the usual VC bounds apply: most deep learning systems have a finite VC dimension
  - We don't have tighter bounds than that.
  - But then again, how many bounds are tight enough to be useful for model selection?
- It's hard to prove anything about deep learning systems
  - Then again, if we only study models for which we can prove things, we wouldn't have speech, handwriting, and visual object recognition systems today

Deep Learning: A Theoretician's Nightmare?

- Deep Learning involves non-convex loss functions
  - With non-convex losses, all bets are off
  - Then again, every speech recognition system ever deployed has used non-convex optimization (GMMs are non convex). hierarchy
- No generalization bounds?
  - Actually, the usual VC bounds apply: most deep learning systems have a finite VC dimension
  - We don't have tighter bounds than that.
  - But then again, how many bounds are tight enough to be useful for model selection?
- It's hard to prove anything about deep learning systems
  - Then again, if we only study models for which we can prove things, we wouldn't have speech, handwriting, and visual object recognition systems today

Deep Learning: A Theoretician's Nightmare?

- Deep Learning involves non-convex loss functions
  - With non-convex losses, all bets are off
  - Then again, every speech recognition system ever deployed has used non-convex optimization (GMMs are non convex). hierarchy
- No generalization bounds?
  - Actually, the usual VC bounds apply: most deep learning systems have a finite VC dimension
  - We don't have tighter bounds than that.
  - But then again, how many bounds are tight enough to be useful for model selection?
- It's hard to prove anything about deep learning systems
  - Then again, if we only study models for which we can prove things, we wouldn't have speech, handwriting, and visual object recognition systems today

Deep Learning: A Theoretician's Nightmare?

- Deep Learning involves non-convex loss functions
  - With non-convex losses, all bets are off
  - Then again, every speech recognition system ever deployed has used non-convex optimization (GMMs are non convex). hierarchy
- No generalization bounds?
  - Actually, the usual VC bounds apply: most deep learning systems have a finite VC dimension
  - We don't have tighter bounds than that.
  - But then again, how many bounds are tight enough to be useful for model selection?
- It's hard to prove anything about deep learning systems
  - Then again, if we only study models for which we can prove things, we wouldn't have speech, handwriting, and visual object recognition systems today

Deep Learning: A Theoretician's Nightmare?

- Deep Learning involves non-convex loss functions
  - With non-convex losses, all bets are off
  - Then again, every speech recognition system ever deployed has used non-convex optimization (GMMs are non convex). hierarchy
- No generalization bounds?
  - Actually, the usual VC bounds apply: most deep learning systems have a finite VC dimension
  - We don't have tighter bounds than that.
  - But then again, how many bounds are tight enough to be useful for model selection?
- It's hard to prove anything about deep learning systems
  - Then again, if we only study models for which we can prove things, we wouldn't have speech, handwriting, and visual object recognition systems today

Deep Learning: A Theoretician's Nightmare?

- Deep Learning involves non-convex loss functions
  - With non-convex losses, all bets are off
  - Then again, every speech recognition system ever deployed has used non-convex optimization (GMMs are non convex). hierarchy
- No generalization bounds?
  - Actually, the usual VC bounds apply: most deep learning systems have a finite VC dimension
  - We don't have tighter bounds than that.
  - But then again, how many bounds are tight enough to be useful for model selection?
- It's hard to prove anything about deep learning systems
  - Then again, if we only study models for which we can prove things, we wouldn't have speech, handwriting, and visual object recognition systems today

Deep Learning Basics

Next Class..